

APPARATUS AND METHOD FOR CONTROLLING A SCREEN POINTER

The Field of the Invention

5 This invention relates generally to devices for controlling a cursor on a display screen, also known as pointing devices. This invention relates more particularly to a touchpad with inertial properties.

Background of the Invention

10 The use of a hand operated pointing device for use with a computer and its display has become almost universal. By far the most popular of the various devices is the conventional (mechanical) mouse, used in conjunction with a cooperating mouse pad. Centrally located within the bottom surface of the mouse is a hole through which a portion of the underside of a rubber-surfaced
15 steel ball extends. The mouse pad is typically a closed cell foam rubber pad covered with a suitable fabric. Low friction pads on the bottom surface of the mouse slide easily over the fabric, but the rubber ball does not skid. Rather, the rubber ball rolls over the fabric as the mouse is moved. Interior to the mouse are rollers, or wheels, that contact the ball at its equator and convert its rotation into
20 electrical signals representing orthogonal components of mouse motion. These electrical signals are coupled to a computer, where software responds to the signals to change by a ΔX and a ΔY the displayed position of a pointer (cursor) in accordance with movement of the mouse. The user moves the mouse as necessary to get the displayed pointer to a desired location or position. Once the
25 pointer on the screen points at an object or location of interest, a button on the mouse is activated with the fingers of the hand holding the mouse. The activation serves as an instruction to take some action, the nature of which is defined by software in the computer.

 A "track ball" is another example of a mechanical type of pointing
30 device. A track ball is essentially an upside-down mouse. In a track ball, rather

than sliding the device itself over a surface to produce pointer movement as in a mouse, a user directly contacts the mechanical ball with the user's finger, and causes the ball to rotate. As with a mouse, the movement of the mechanical ball in a track ball generates a corresponding movement of the displayed pointer.

5 In a track ball, the mechanical ball can be "flicked" with the finger, and the ball will continue to rotate under its own momentum after the user's finger is removed from the ball. The rotation continues until the user contacts the mechanical ball again, or until frictional forces eventually cause the ball to stop rotating. The inertial properties of a track ball that allow it to continue to
10 generate pointer movement after the user stops contacting the mechanical ball result in good dynamic range. Small hand movements can result in large pointer movements. The inertial properties of a track ball are useful in some applications, such as game applications, where large and quick pointer movements are sometimes desirable. However, trackballs are not typically
15 incorporated into modern laptop computers because such computers are designed to use more planar pointing devices, and the height of a typical trackball makes it incompatible with most laptop computers.

 Some mechanical mouse devices may also provide inertial effects like a track ball. A mechanical mouse may be moved quickly over the mouse pad, and
20 then lifted from the pad, allowing the ball to continue to rotate under its own momentum. Some mechanical mouse devices, however, cause the ball to immediately stop movement when the mouse is lifted from the mouse pad.

 Conventional touchpads found on many laptop computers do not use a mechanical ball, or other similar moving mechanical element that has inertial
25 properties. Touchpads are essentially writing surfaces that capture the position of a pointing object, such as a finger, pen, or stylus, which is moved across the writing surface. With a touchpad, position information is typically determined by sensing changes in resistance or capacitance. Touchpads are incorporated into many modern laptop computers because, unlike a trackball, touchpads are
30 more planar, and have a size that permits easier incorporation into a laptop computer. Unfortunately, existing touchpads do not have much finesse, and

several swipes across the touchpad surface must typically be performed to move the screen pointer to a desired location. It would be desirable in some applications for a touchpad to provide inertial effects, such as that provided by a track ball.

5

Summary of the Invention

One form of the present invention provides an apparatus for controlling a position of a screen pointer for an electronic device having a display screen. The apparatus includes a touchpad including a touch-sensitive surface on which a pointing object may be placed. The touchpad is configured to generate position signals indicative of positions of the pointing object on the touch-sensitive surface. The apparatus includes a controller for generating a first set of movement data based on the position signals. The first set of movement data is indicative of motion of the pointing object across the touch-sensitive surface. The controller is configured to generate a second set of movement data when the pointing object is removed from the touch-sensitive surface, thereby leaving the touch-sensitive surface free from contact by a pointing object. The second set of movement data is indicative of motion of the pointing object across the touch-sensitive surface prior to removal of the pointing object.

20

Brief Description of the Drawings

Figure 1 is a diagram illustrating a laptop computer with a touchpad according to one embodiment of the present invention.

Figure 2 is a block diagram illustrating major components of the laptop computer shown in Figure 1 according to one embodiment of the present invention.

25

Figure 3 is a graph illustrating typical velocity profiles for a mechanical pointing device, such as a track ball, and a touchpad.

Figure 4 is a block diagram illustrating one embodiment of a momentum simulator or emulator for a touchpad that provides inertial effects according to one embodiment of the present invention.

30

Figure 5 is a flow diagram illustrating a process for providing inertial effects in a touchpad according to the present invention.

Description of the Preferred Embodiments

5 In the following detailed description of the preferred embodiments, reference is made to the accompanying drawings, which form a part hereof, and in which is shown by way of illustration specific embodiments in which the invention may be practiced. It is to be understood that other embodiments may be utilized and structural or logical changes may be made without departing
10 from the scope of the present invention. The following detailed description, therefore, is not to be taken in a limiting sense, and the scope of the present invention is defined by the appended claims.

Figure 1 is a diagram illustrating a laptop computer 1 with a touchpad 4 according to one embodiment of the present invention. Computer 1 includes
15 display 2, keyboard 3, touchpad 4, and buttons 6. Touchpad 4 includes a rectangular-shaped touch-sensitive surface 5. Touchpad 4 is configured to track the movement of a human finger or other pointing object pressed against the surface 5 of the touchpad 4. In one embodiment, touchpad 4 is a resistive type touchpad that identifies the position of a finger pressed against the surface 5
20 based on sensed changes in resistance. In another embodiment, touchpad 4 is a capacitive type touchpad that identifies the position of a finger pressed against the surface 5 based on sensed changes in capacitance. Computer 1 is described in further detail below with reference to Figure 2.

Figure 2 is a block diagram illustrating major components of the laptop
25 computer 1 shown in Figure 1 according to one embodiment of the present invention. Computer 1 includes processor 10, video adapter 11, display 2, system bus 12, memory 17, serial interface 16, keyboard 3, and touchpad 4.

Processor 10 communicates with various components of computer 1, including video adapter 11, serial interface 16, and memory 17, via system bus
30 12. Video adapter 11 is coupled to display 2, and drives display 2 under the control of processor 10. Touchpad 4 and keyboard 3 are communicatively

coupled to processor 10 via serial interface 16 and system bus 12. Motion data received from touchpad 4 is passed from serial interface 16 to processor 10 for processing. Memory 17 includes operating system 13 and one or more application programs 15. Operating system 13 includes touchpad driver 14,
5 which is used by processor 10 in processing motion data received by touchpad 4.

Although one embodiment of the present invention is described in the context of a laptop computer system, the techniques described herein are applicable to any type of electronic device with a touchpad, including, but not limited to, a cellular telephone, personal digital assistant (PDA), portable music
10 player (e.g., MP3 player), pager, portable game device, or other device.

Touchpad 4 generates electrical signals indicative of positions of a finger or other pointing object against surface 5 of touchpad 4. In one form of the invention, the electrical signals are generated based on sensed changes in impedance (e.g., resistance or capacitance) caused by the touching of surface 5
15 by a pointing object. In one embodiment, touchpad 4 includes a controller 18 that generates digital ΔX and a ΔY relative motion data based on the electrical signals. In another embodiment, processor 10 generates ΔX and a ΔY relative motion data based on position signals or data generated by touchpad 4. In one form of the invention, the motion data generated by touchpad 4 are provided
20 through serial interface 16 to processor 10. Processor 10 responds to the motion data received from touchpad 4 to change by a ΔX and a ΔY the displayed position of a pointer (cursor) on display 2. A user moves his finger as necessary against surface 5 of touchpad 4 to get the displayed pointer to a desired location or position on display 2. Once the pointer on the display 2 points at an object or
25 location of interest, one of the buttons 6 is activated by the user. The activation serves as an instruction to processor 10 to take some action, the nature of which is defined by software stored in memory 17, such as driver 14 or application program 15. In another embodiment, the activation is provided by tapping on the surface 5 of touchpad 4. Processor 10 monitors the current position of the
30 screen pointer displayed on display 2 to take appropriate action based on the

position of the screen pointer when a user activates a button 6 or taps on the surface 5 of touchpad 4.

If a fingertip was previously touching surface 5, but is then lifted away from the surface 5, this condition is detected by touchpad 4, and in one
5 embodiment, the production of incremental (X, Y) signals continues, with gradually decreasing magnitudes. This has the effect of continuing the motion of the screen pointer. In one form of the present invention, the continued motion is similar to the motion of a screen pointer when a mechanical ball of a track ball is "flicked." When the fingertip is subsequently replaced on surface 5, touchpad 4
10 detects this condition and treats the situation as though a reset had been performed. That is, until there has been new motion subsequent to the new placement of the fingertip on surface 5, the incremental coordinates (X, Y) will have the value (0, 0). Thus, if the screen pointer is moving when the fingertip is replaced on surface 5, the movement stops when touchpad 4 detects the contact.
15 The screen pointer remains at the position where it stopped until touchpad 4 detects new motion of the fingertip on surface 5.

Figure 3 is a graph illustrating typical velocity profiles for a mechanical pointing device, such as a track ball, and a touchpad. Graph 20 includes a
20 horizontal axis, a vertical axis, curve 24, and curve 26. The horizontal axis of graph 20 represents time, and the vertical axis represents velocity. Curve 26 illustrates the velocity versus time or velocity profile of a mechanical mouse or track ball when it is "flicked", or quickly accelerated and then released. The total distance moved is proportional to the area under curve 26. Curve 26 is generated from a series of velocity samples 22. As shown by curve 26, the
25 velocity of the mechanical ball in the track ball initially increases at a near constant rate, with the velocity increase caused by a finger beginning a flicking action on the ball. After the mechanical ball is flicked and released, the velocity of the ball gradually decays to zero. In contrast, for a typical touchpad, when a user removes his finger from the touchpad after performing a flicking action, the
30 velocity instantly drops to zero as shown by curve 24. Curves 24 and 26 are representative of typical velocity profiles for one type of pointer device

movement, and will vary depending upon the exact movement caused by a user. In one form of the present invention, the velocity profile of a mechanical mouse or track ball represented by curve 26 is used as a model in determining an appropriate velocity profile for a touchpad with inertial properties.

5 Figure 4 is a block diagram illustrating one embodiment of a momentum simulator or emulator 40 for a touchpad that provides inertial effects according to one embodiment of the present invention. Momentum simulator 40 includes friction simulator or controller 44, velocity state variable register 46, and time
10 constant 48. In one embodiment, momentum simulator 40 and controller 18 are integrated into a single IC package. In another embodiment, momentum simulator 40 and controller 18 are separate ICs. Controller 18 outputs velocity data for X and Y dimensions as discussed above with reference to Figure 2. In one embodiment, controller 18 multiplies velocity data by a multiplication factor, thereby normalizing the velocity data. The normalized velocity data is
15 output to friction simulator 44, which, in effect, adds friction to the velocity data, and outputs gradually reduced velocity data to processor 10.

 In one embodiment, the rate at which the velocity data is reduced by friction simulator 44 is based on an exponential decay with a time constant specified by the value of time constant 48. In one form of the invention, the
20 value of time constant 48 may be specified by a user. Friction simulator 44 stores velocity data in velocity state variable register 46. Momentum simulator 40 is described in further detail below with reference to Figure 5.

 Figure 5 is a flow diagram illustrating a process 60 for providing inertial effects in a touchpad 4 according to one embodiment of the present invention. In
25 step 62, controller 18 determines whether a fingertip or other pointing object is touching surface 5. If a fingertip is not touching surface 5, controller 18 outputs zero velocity, and continues to sense whether a fingertip touches surface 5. If a fingertip is touching surface 5, controller 18 calculates the X and Y components of the velocity of movement of the fingertip, and, in step 64, outputs the velocity
30 data or movement data to friction simulator 44. In step 66, controller 18 senses whether the fingertip has been removed from surface 5. If the fingertip has not

let go of surface 5, controller 18 continues to output velocity data as represented by step 64. Thus, as long as the fingertip continues to move across surface 5, controller 18 continues to output velocity data representative of the movement. The velocity data output by controller 18 is normalized. In one embodiment, controller 18 multiplies the velocity data by 100, and outputs the normalized velocity data to friction simulator 44. Friction simulator 44 outputs the normalized velocity data without adding friction, since the fingertip is still on surface 5.

In step 68, when the fingertip is removed from surface 5, the velocity at the time of removal, or the current velocity data, is stored in register 46. In step 70, friction simulator 44 decreases the velocity data in cycles by an amount determined by time constant 48. In step 74, friction simulator 44 stores the reduced velocity data for the current cycle in register 46. In step 76, friction simulator 44 determines whether the reduced velocity data stored in register 46 is below a specified threshold level. If the reduced velocity data is below the specified threshold level, friction simulator 44 outputs zero velocity in step 80, and controller 18 again waits for the fingertip to touch surface 5 as indicated by step 62. If the reduced velocity data stored in register 46 is not below the specified threshold level, in step 78, friction simulator 44 outputs the reduced velocity data. If the fingertip does not re-touch surface 5 (step 72), friction simulator 44 continues to gradually reduce the velocity data and output the reduced velocity data (steps 70, 74, 76, and 78) until the velocity data is below the specified threshold level. At step 72, if the fingertip re-touches surface 5, the process jumps to step 64, where controller 18 determines the current velocity of movement and outputs corresponding velocity data.

To further illustrate the process 60 according to one embodiment of the present invention, an example will be provided. Assume that the time constant 48 is one second, and that momentum simulator 40 outputs velocity data at 100 samples/second. These values are chosen to simplify the mathematics, and other values may be more appropriate for a commercial device. Assume further that a fingertip has just been removed from surface 5, and that the normalized velocity

output by controller 18 at the time of removal is 1.0 inches/second. Friction simulator 44 stores the normalized velocity in register 46. During the first cycle period (i.e., the first millisecond), friction simulator 44 decreases the velocity stored in register 46 by 1 percent, stores the decreased velocity (i.e., 0.9900 inches/second) in register 46, and outputs the decreased velocity. During the second cycle period, friction simulator 44 again decreases the velocity stored in register 46 by 1 percent, stores the decreased velocity (i.e., 0.9801 inches/second) in register 46, and outputs the decreased velocity. Friction simulator 44 continues to decrease the velocity by 1 percent each millisecond until the velocity is below the specified threshold value, at which point friction simulator 44 outputs a value of zero velocity.

For a time constant 48 of 1 second, after the first second, friction simulator 44 drives the initial velocity of 1 inch/second to 0.37 inches/second. After two seconds, friction simulator 44 drives the velocity down to 0.14 inches/second. Assuming that the specified threshold value is 0.10 inches/second, friction simulator 44 drives the initial velocity down to zero just after 2 seconds. Other values for time constant 48 may be chosen to provide a desired level of friction. In addition, acceleration and deceleration characteristics may be explicitly defined by a user. In one embodiment, faster flicks will result in larger initial velocities and longer settling times, and slower flicks will result in smaller initial velocities and shorter settling times. In one embodiment, the inertial motion may be stopped at any time by re-touching surface 5. In one embodiment, the continued inertial movement of the present invention is provided only when a specified minimum acceleration has been provided by a user.

Rather than using a velocity profile based on exponential decay as discussed above, alternative techniques may be used. For example, in one embodiment, a user can graphically enter a sample velocity profile, and a mathematical representation of the entered profile is automatically generated. Friction simulator 44 then uses the mathematical representation to generate the desired velocity characteristics.

In another embodiment, a user can enter a “flick factor”, which defines the screen pointer movement when the touchpad 4 is flicked. For example, a 1 inch/second flick of the touchpad 4 could cause the screen pointer to traverse the screen and slide to a stop.

5 Those of ordinary skill in the art will recognize that there are numerous ways for a user to enter data representing desired pointer movement characteristics. The entered data can then be used by friction simulator 44 to generate the desired response. For example, in one embodiment, a user may specify that the device is to have no friction, so that when the touchpad 4 is
10 flicked, the pointer will continue across the screen until it reaches a screen boundary, or until the user again contacts the touchpad 4. In another embodiment, a user may specify that the touchpad 4 is to provide further acceleration of the screen pointer after a user flicks the touchpad 4. Such further acceleration could not be provided with a simple mechanical track ball, due to
15 the laws of physics. When a track ball has been released, it cannot accelerate further unless pushed by a user. Any mathematical function or arbitrary curve may be entered by a user to specify the desired response of the touchpad 4, including functions or curves that eventually result in zero velocity, and functions and curves that result in continued velocity and/or acceleration after a
20 user stops contacting the touchpad 4. The continued velocity and/or acceleration can be overridden by the user by contacting the touchpad 4. A user may exactly specify the desired characteristics of the touchpad 4 for each particular application.

 It will be understood by a person of ordinary skill in the art that functions
25 performed by controller 18 and momentum simulator 40 may be implemented in hardware, software, firmware, or any combination thereof. The implementation may be via a microprocessor, programmable logic device, or state machine. Components of the present invention may reside in software on one or more computer-readable mediums. The term computer-readable medium as used
30 herein is defined to include any kind of memory, volatile or non-volatile, such as

floppy disks, hard disks, CD-ROMs, flash memory, read-only memory (ROM), and random access memory.

As one example, rather than implementing the inertial properties or other movement characteristics in a sensor or other chip (e.g., momentum simulator
5 40) as part of touchpad 4, the desired movement characteristics may be implemented in a software driver of a host device (e.g., computer, PDA, digital camera, cell phone, portable game device, etc.) coupled to the touchpad 4.

Embodiments of the present invention provide numerous benefits, including the ability to increase the dynamic range of a touchpad, which is
10 particularly useful for larger screens. Small hand movements can result in large screen pointer movements. Touchpad 4 according to one embodiment provides a large dynamic range similar to a trackball, but with a shallower depth than a typical trackball, thereby making it easier to incorporate touchpad 4 into a laptop computer than a typical trackball. Also, a touchpad 4 like that shown in Figure 1
15 can be made to feel and operate just like a mechanical track ball, which may be desirable to individuals who are familiar with operating such mechanical pointing devices.

Although specific embodiments have been illustrated and described herein for purposes of description of the preferred embodiment, it will be
20 appreciated by those of ordinary skill in the art that a wide variety of alternate and/or equivalent implementations may be substituted for the specific embodiments shown and described without departing from the scope of the present invention. Those with skill in the mechanical, electro-mechanical, electrical, and computer arts will readily appreciate that the present invention
25 may be implemented in a very wide variety of embodiments. This application is intended to cover any adaptations or variations of the preferred embodiments discussed herein. Therefore, it is manifestly intended that this invention be limited only by the claims and the equivalents thereof.